PTO 03-4961 German

Document No. DE 195 48 387 C1

Process for Cryptographic Securing of Computer-Assisted Digital Communication Between a Program and at Least One User Unit

[Verfahren zur kryptographischen Sicherung der rechnergestützten digitalen Kommunikation zwischen einem Programm und mindestens einer Benutzereinheit]

Dr. Oliver Pfaff

UNITED STATES PATENT AND TRADEMARK OFFICE

Washington, D.C. August 2003

Translated by: Schreiber Translations, Inc.

Country	:	Germany
Document No.	:	DE 195 48 387 C1
Document Type	:	Patent Application Laid Open to Inspection
Language	:	German
Inventor	:	Dr. Oliver Pfaff
Applicant	:	Siemens AG
IPC	:	H 04 L 9/00, G 06 F 12/14, G 06 F 13/42
Application Date	:	December 22, 1995
Publication Date	:	January 30, 1997
Foreign Language Title	:	Verfahren zur kryptographischen Sicherung der rechnergestützten digitalen Kommunikation zwischen einem Programm und mindestens einer Benutzereinheit
English Language Title	:	Process for Cryptographic Securing of Computer-Assisted Digital Communication Between a Program and at Least One User Unit

/1[1]

Specification

This invention relates to a process for the cryptographic securing of communication in so-called Client-Server window systems with an open network interface. An example of such Client-Server window systems is described in [1].

By additionally using a so-called Application Sharing Component, which is used to multiplex the requests of user units addressed to the program and to demultiplex communications from the program, for example, event messages, responses or error messages, one can achieve the common use of a Standard Single-User Application environments that can be located in different places.

The common processing of a program is referred to as Application Sharing.

But to achieve reliable communication of confidential data, the Client-Server window system, described in [1], must be expanded by cryptographic characteristics.

This is particularly important in communication between various enterprises. Here is what that means: In case of a communication between computers where one computer is in the basically secured, confidence-worthy, so-called Corporate

---

[1] Numbers in the margin indicate pagination in the foreign text.

Network of an enterprise and other computers that [are] in a common synchronously distributed working environment of several cross-linked computers in a so-called Computer System Cooperated Work System (CWCW systems) that implements Application Sharing, can be achieved only via an unsecured channel, which means that secure communication is no longer guaranteed.

Such CSCW systems are based on the possibility of processing a standard single-user application (Standard Single User Application) together with other units at one particular point in time.

The information exchanged between the computers can be of special significance, for example, it may involve confidential business information, design specifications, financial transactions or medical data that are exchanged via the unsecured channel.

This is why it is necessary to ensure a certain degree of security also for these transactions that involve application data.

In many commercial systems that are based on the Client-Service window system described in [1], direct integration of cryptographic characteristics is not possible.

The problem behind the invention thus is to provide a process for the cryptographic securing of computer-assisted

digital communication between a program and at least one user unit.

The problem is solved by the process according to Claim 1, the process according to Claim 4, the process according to Claim 7 as well as the process according to Claim 8.

In the process according to Claim 1, the program forms a communication that is coded for at transport protocol. Directly after encoding using the transport protocol, the encoded communication is again decoded and the decoded communication is subjected to the cryptographic process. Then the request is again encoded with the transport protocol and is transmitted to at least one user unit. In this case, the program and the user unit can be in one or also on various computers.

In the process according to Claim 4, one basically takes the same steps with the difference that this time a request is formed in a user unit and that the additional steps described above are also taken there. To complete the procedure, the encoded cryptographically processed request is transmitted to the program in this case.

In the process according to Claim 7, one starts with the following situation: On the side of the program, an expansion of the Client-Server window system is possible by means of security mechanisms of the most varied kind, which will be described below. For this case, the above-described process

steps will be performed in a user unit starting with the formation of a communication in the program only after reception of the requests that were cryptographically processed in the inserted security layer on the side of the program. In other words, the inverse cryptographic processes are performed there in order to process the communications and this process step is characterized by a prior decoding with the transport protocol and subsequent encoding with the transport protocol.

The process according to Claim 8 basically features the same process steps as the process according to Claim 7 with the following difference: Here a request is formulated by a user unit and transferred to the program. The placed where the individual process steps occur and the places where the process steps of Claim 7 are performed are simply switched around in this case.

Advantageous developments of the invention-based process will result from the subclaims.

It is advantageous by way of cryptographic processing to provide at least encoding of the request. That ensures the confidentiality of the exchanged data.

It is also advantageous by way of cryptographic processing of the request to provide integrity and authentication mechanisms; this means that one can in each case guarantee then

that the received communication actually comes from the sender, who is also listed as sender in the communication.

As a further development of the invention-based process, it is also advantageous to provide access control mechanisms as cryptographic processes in order thus to make sure that really only those requests are carried out which also have the entitlement for implementation.

It is also advantageous prior to the start of the process during an initialization phase, for example, to exchange between the program and at least one user unit the cryptographic codes that are employed to carry out the individual cryptographic processes.

The processes are used advantageously during data exchange between communication partners that [takes place] across the boundaries of a Corporate Network, which is secured by means of cryptographic processes via an unsecured channel in a so-called Firewall. /2

By virtue of this manner of use, it is no longer necessary as in the past in handling an intended communication beyond Corporate Network boundaries to uncouple the computer used for the communication from the entire network of the Corporate Network in order thus not to endanger the entire Corporate Network in case of possible attacks via the unsecured communication channel.

The figures show some exemplary embodiments, which will be explained in greater detail below.

Fig. 1 shows the general principle of a Client-Server window system;

Fig. 2 shows the general principle of a Client-Server window system in a "multiuser environment";

Fig. 3 shows an arrangement that describes the multiuser environment in a more detailed fashion;

Fig. 4 is a basic block diagram describing the insertion of a security layer between the Client-Server window system and the transport protocol;

Fig. 5 shows an arrangement basically illustrating how the invention-based process can be used in a Firewall to secure the communication beyond Corporate Network boundaries;

Fig. 6 is a flow chart illustrating the process steps of the process according to Claim 1;

Fig. 7 is a flow chart illustrating the process steps of the process according to Claim 2;

Fig. 8 is a block diagram describing the individual possibilities for carrying out the security-specific processing of the request or the inverse security-specific processing of the request;

Fig. 9 is a flow chart illustrating the individual process steps of the process according to Claim 4;

Fig. 10 is a flow chart illustrating the steps of the process according to Claim 5;

Fig. 11 is a flow chart illustrating the steps of the process according to Claim 7;

Fig. 12 is a flow chart illustrating the steps of the process according to Claim 8;

Fig. 13 is a block diagram describing the individual components needed to implement the process according to Claim 1 and the communication exchange [procedure];

Fig. 14 is a block diagram describing the individual components needed to implement the process according to Claim 4 and the communication exchange [procedure];

Fig. 15 is a block diagram describing the individual components needed to implement the process according to Claim 2 and the communication exchange [procedure];

Fig. 16 is a block diagram describing the individual components needed to implement the process according to Claim 5 and the communication exchange [procedure];

Fig. 17 is a block diagram describing the individual components needed to implement the process according to Claim 7 and the communication exchange [procedure];

Fig. 18 is a block diagram describing the individual components needed to implement the process according to Claim 8 and the communication exchange [procedure].

The invention will be explained in greater detail with reference to Claims 1 to 18.

Fig. 1 shows a user environment that occurs, for example, in a Client-Server window system that is described in [1].

This arrangement displays at least the following components:

- a user unit XS, hereafter also referred to as Server XS, which again has the following components:

- at least one driver unit DD which facilitates coupling between additional peripheral components with a client XC, described below,

- a display screen unit BS,

- a keyboard TA,

- a mouse MA,

- the client XC that displays at least the following components:

- a quantity of library routines XL as well as

- an application ANW.

Display screen unit BS, keyboard TA, mouse MA and possibly other peripheral units can be present additionally along with the peripheral components described above, which are coupled to client XC via the corresponding driver units DD.

The quantity of the library routines XL of client XC forms the interface between application ANW, for example, a text

processing program or also a table calculation program or all other known applications ANW and the user unit XS.

Together, the library routines XL as well as the application ANW form one program P.

Although in this exemplary embodiment we describe in each case only one application ANW or one program P, one naturally can supply several applications ANW and thus several clients XC on one computer unit that performs this particular application ANW.

This request, illustrated in Fig. 1, is thus only a very simple, basic example for the routing of the communication of a client XC with the server XS such as it is carried out via the known Client-Server window system described in [1].

Server XS transmits a request A to client XC. As a result, actions are triggered in client XC, for example, in application ANW.

Request A, for example, can be an input on keyboard TA that via the driver units DD is "translated" into request A and is transmitted to client XC.

Application ANW, for example, a text processing program or a calculation program, a symbol program and similar programs can now accept the input and, for example, include a new letter in the text data file.

To make sure that this change in the text data file can also be illustrated on the display screen BS in a response B, in this case, for example, an illustration communication, is transmitted to display screen unit BS by means of which a change in the display screen illustrated is requested. /3

One disadvantage inherent in many commercial systems that work according to this principle resides above all in the fact that a direct integration of needed security mechanisms into the Client-Server window system is often impossible.

That, it so happens, would require direct interrupt into the interface between the library routines XL and the transport protocols. The latter, it just so happens, are often not accessible to the user.

Fig. 6 shows a flow chart with individual process steps involved in the invention-based process according to Claim 1. The arrangement, needed to implement this process, is described in Fig. 13.

Program P formulates communication B in a first step 601.

A new communication is formed from communication B in a transport protocol layer TP in that communication B is "embedded" into the transport protocol format, in other words, it is coded 602, CB.

An overview of the various transport protocols can be found in [2]. The invention-based processes are independent of the special particular transport protocol that is being used.

Either on the same computer unit on which program P runs or on a separately provided first security computer unit SC1, which is coupled to the computer via a secure channel, the coded communication CB is decoded 603, DB in the transport protocol layer TP that is provided there.

The decoded communication DB is now routed to a security layer SL in which it is subjected 604 to various randomly predetermined cryptographic procedures.

A cryptographically processed communication VB, formed by cryptographic processing, is now again encoded 605 in the transport protocol layer TP, as a result of which, a coded cryptographically processed communication CVB is formed.

The coded cryptographically processed communication CVB is transferred in a last step 606 to the user unit XS, in other words, to the Server.

The basically reverse case for request A from Fig. 1 is illustrated in Fig. 9 in the form of a flow chart and in Fig. 14 in the form of a block diagram for the arrangement that is needed to implement the process.

In this case, request A is formed 901 by the user unit XS.

Request A is returned to the transport protocol layer TP and it is there embedded 902 into the particular transport protocol format that is used. An encoded request CA, resulting from that, is now - either in user unit XS itself or in a separately provided second secure computer unit SC2 that is coupled with user unit XS via a secure channel - "unpacked" in transport protocol layer TP, in other words, it is decoded 903, which forms a decoded request DA.

In security layer SL, the decoded request DA that is supplied to it will now be subjected 904 to the provided cryptographic processes that will be described below. This results in a cryptographically processed request VA.

The cryptographically processed request VA again is supplied 905 to the transport protocol unit TP and it is encoded there, as a result of which, there is formed an encoded cryptographically processed request CVA. The encoded cryptographically processed request CVA is transmitted in a last step 906 to the program P, in other words, to the Client XC.

A development of the process according to Claim 1 is shown in Fig. 7 in the form of a flow chart and the arrangement needed to implement this process is shown in Fig. 16.

After the performance of the process steps shown in Fig. 6 to the lastly formed encoded cryptographically processed communication CVB that is transmitted to the user unit XS, the

encoded cryptographically processed communication CVB is received 701 by the at least one user unit XS or by the second secure computer unit SC2.

Using the transport protocol employed for encoding, the encoded cryptographically processed communication CVB is "unpacked," in other words, it is decoded 702 in the transport protocol layer TP of the user unit or of the second security computer unit SC2.

That forms a decoded cryptographically processed communication DVB that is now supplied to the security layer SL, which is also provided on the side of the user unit XS or the second security computer unit SC2. In the security layer SL, the decoded cryptographically processed communication DVB is subjected 703 to the particular inverse cryptographic processes. In this context, inverse means inversely with respect to the cryptographic processes that were applied in the security layer of the Client XC or of the first security computer unit SC1 upon the decoded communication DB.

The result of this cryptographic processing is an inversely cryptographically processed communication DEB, which now again is supplied to the transport protocol layer TB, where it is also again encoded 704.

The resultant encoded inversely cryptographically processed communication CEB is again supplied to the transport protocol layer TB and is decoded 705 there.

The resultant communication is now supplied to the actual Server XS, in other words, to the user unit XS, and is further processed there. A variant of the process is of course also possible in terms of directly further processing the inversely cryptographically processed communication DEB.

The basically identical development of the process according to Claim 4, that is, the same as in the previously described development for the process according to Claim 1, is illustrated in Fig. 10 along with the arrangement needed to implement the process according to Claim 5 shown in Fig. 17.

In this development again, one starts with the idea that the process steps, described in Fig. 9 up to the encoding of the cryptographically processed request VA and their transmission to the program P, have been carried out.

The transmitted encoded cryptographically processed requirement CVA is received 1001 by the program P or by the first security computer unit SC1.

In a further step 1002 using the transport protocol, one again "unpacks" the encoded cryptographically processed request CVA, in other words, it is decoded in the transport protocol layer TP.

/4

Furthermore, the resultant decoded cryptographically processed request DVA is subjected 1003 to the cryptographic processing that is inverse with respect to the employed cryptographic process in the security layer SL to which it was supplied.

The resultant inversely cryptographically processed request DEA is now again coded 1004 in the transport protocol layer TP.

Then it is again decoded 1005 in the transport protocol layer TP and is supplied to program P. That is where the actual request A is further processed.

Again it is just as well possible directly to supply the decoded inversely cryptographically processed request DEA to the program P and to process it further there.

Fig. 11 describes another process that is similarly based on the common inventive idea of the processes described above.

This time, however, it is presumed that it is possible directly to insert a security layer SL between Client XC and transport protocol layer TP. This now no longer creates the need on the side of Client XC twice "to run through" the transport protocol layer TP.

This is illustrated in the arrangement in Fig. 17.

Here again, program P formulates 1101 the communication B. But communication B, however, this time is directly subjected VB, 1102 to a cryptographic process in security layer S- [sic].

The resultant cryptographically processed communication VB is supplied to transport protocol layer TP where it is encoded 1103.

The encoded cryptographically processed communication CVB is transferred 1104 to the user unit XS, there it is received 1105 by user unit XS or by the second security computer unit SC2, it is decoded in the transport protocol layer TP that is provided there into the decoded cryptographically processed communication DVB 1106.

The latter is supplied to security layer SL and is there subjected 1107 to the inverse cryptographic process or processes.

In the last two steps, the inversely cryptographically processed communication DEB is again encoded 1108 in transport protocol layer TP and is decoded 1109 in a last step.

The resultant communication B is supplied to Server XS and is further processed.

Security layer SL is illustrated in Fig. 4 for the case where it is possible to insert the security layer SL between the transport layer TP and the library routines XL.

This time, however, looking at the special example which in no way whatsoever restricts the general validity, unsecured read, write, readv, writev connect and accept communications are "secured" by the cryptographic process provided in security

layer SL. This is done by applying the provided cryptographic processes upon the particular communication B or request A. The communications that are "secured" by security layer SL are marked in Fig. 4 by an asterisk *.

The described cryptographic securing of the communication of an application with a window system via a network, on the one hand, presupposes the exchange of the cryptographic codes and, on the other hand, is based on a reciprocal authentication of the two communications partners.

For this authentication, one can advantageously employ asymmetrical cryptographic processes together with certificates that contain public codes. By suitable definition of the identity characteristics in the certificate, it is possible to identify and authenticate services such as applications or window service programs beyond the mere computer address in the network. Such identity characteristics that go beyond the network address for the differentiation of various application programs of a computer, for example, can be the name of the service owner on a multiuser system.

The reciprocal authentication and the code exchange are carried out in an initialization phase to build up the secure connection.

As a further development of the invention-based process, it is advantageous on the side of the window service program, in

other words, the user unit XC, to carry out an access check on the basis of the authenticated identity of program P. The authenticated identification information can go beyond the computer address of program P; therefore, an access check can differentiate between various programs P of a computer and can thus control the buildup of the connection.

An advantageous application of the described security procedure can be found in the exchange of application data between a program P and a window service program, in other words, a user unit XC, where only one network connection that is not worthy of confidence can be switched between both of them.

This scenario is especially important to the above-described CSCW systems that do the Application Sharing. Here the participating window service programs of the user units XC are often in different company networks and can exchange data with the application or the Application Sharing component only via public networks.

Considerable security problems are connected with the operation of the known window system and they are described in [6], [7]. Due to the considerable risk potential that is tied to the window system known from [1], the operators of company networks as a rule do not allow such window service programs to cooperate with applications outside the company network. This is intended to predict internal company information and data

inventories. This protection is provided by the so-called Firewalls at the network transition between the in-house network and the outside networks. By filtering data packets on the transport system level, those outside networks prevent external application programs from accessing the in-house window service programs.

These customary precautions, however, prevent the use of synchronous CSCW systems that are based on the following: Users at various sites and in various companies together cooperate via a synchronous CWCW system and together work with application programs.

On the basis of the described security process for application data, one can construct a program for a Firewall, which makes it possible to allow in-house window service programs securely to communicate with outside application programs:

This special program is based, on the one hand, on the described security expansion to protect application data in window systems and, on the other hand, on a gating component for application data. The gating component can be derived directly from the Application Sharing Component ASC because in this case there is no request for multiplexing and demultiplexing. /5

These two components (security service program and gating component) form a special Firewall security service program by

means of which it becomes possible from an external application program to request a specific authentication as well as to subject it to an access check before the gating component establishes the connection to the in-house window service program and then switches the connection through. The subsequent data exchange between the external application program and the Firewall security service program is protected by cryptographic mechanisms.

By operating packet filters in the Firewall, one can force external application programs first of all to establish connection with the described security service program.

The corresponding process, considering the "switch of roles" between program and user unit XS, in other words, for request A, is illustrated in Fig. 12 along with the arrangement needed for its implementation shown in Fig. 18.

Here one naturally assumes that security layer SL can be inserted on the side of the user unit XS between the user unit XS and the transport protocol layer TP.

On the basis of this assumption, user unit XS thus forms 1201 the request A. This request is directly subjected 1201 to the cryptographic process VA in security layer SL.

The cryptographically processed request VA is encoded 1203 in the transport protocol layer TP and subsequently thereto the

encoded cryptographically processed request CVA is transmitted 1204 to the program P.

There it is received 1205 by program P or by the first security computer unit SC1. In a transport protocol layer TP, which is also provided there, it is now decoded 1206 to the decoded cryptographically processed request DVA.

In security layer SL to which it is supplied in an additional step, the decoded cryptographic request is subjected 1207 to the inverse cryptographic process. The resultant inversely cryptographically processed request DEA is again "packed up" in the transport protocol layer TP, in other words, it is encoded 1208.

The encoded inversely cryptographically processed request CEA is again decoded 1209 in an additional step in transport protocol layer TP and the resultant request A that is now cryptographically "secured" is supplied to the program and is further used by the program P.

Various possibilities for implementing the cryptographic processes to be used in security layer SL are illustrated in Fig. 8.

First of all, it is possible to apply encoding processes 81 in security layer SL. In that way, one can achieve a confidentiality or integrity of the exchanged communications B or requests A.

It is furthermore provided that authentication mechanisms 82 can also be used in security layer SL. These mechanisms make it possible to verify identity data of the communications partners in the network. These authentication mechanisms have a special meaning in the context, for example, of the Transport Control Protocols (TCP) or also the User Datagramm Protocols (UDP) because they do not display any authentication mechanisms for senders and recipients.

The implementation of access check mechanisms 83 that are based on the authentication processes also offers additional protection for the access to the window service program in a Client-Server window system.

The processes described above naturally can also very advantageously be applied to multiuser systems.

The way in which the Client-Server window system described in [1] can be expanded into a multiuser system is described, for example, in [3], [4], [5].

The resultant situation with an additional multiplexer component ASC and several user units XSi, where an index I definitely identifies each user unit XSi and is a natural number in the range from 1 to n, is illustrated in Fig. 2.

Here, the requests Ai are in the known manner combined by the individual user units XSi and communication B is transmitted to the individual user units XSi as copies of communication Bi.

The invention-based process in this context naturally are performed for each individual connection between client XC and each user unit XSi.

This "multiuser environment" is described in even greater detail in Fig. 3. In this practical implementation, the requests Ai correspond to so-called Xrequests and the communications Bi correspond to the so-called Xreplies, Xevents, Xerrors. The application ANW accesses the system resources SR via the system calls SC.

By way of a further development of the process, it is advantageous at the start of the process to provide for an initialization phase in which, for example, one performs a code exchange as well as a bilateral authentication between a user unit XS of the user units XSi and the program P.

The most varied processes for code exchange are known in this case to the expert. As an example of an initialization phase that can be employed in the invention-based process, we propose the following procedure:

The process for code exchange, described below, is generally carried out between client XC and a user unit XS. The multiplexer component ASC in this context is to be considered as a special component of client XC.

Assuming that the multiplexer component ASC has an application certificate and that the user units, in other words,

the servers XSi in each case do have a user certificate which in each case unambiguously are associated with the user units, the multiplexer component ASC then generates a first random number.

After a transport connection between the multiplexer component ASC and the particular server XSi has been filled up, the multiplexer component ASC transmits a first negotiation communication to the user unit, which at least displays the following components: /6

- the program certificate,
- the first random number,
- a first proposal for a cryptographic process that is to be used further on and
- a digital signature that is formed at least via the first random number as well as the first proposal.

The first negotiation communication is received by the particular user unit, in other words, the server XSi.

The user unit XSi checks the program certificate for correctness.

The digital signature is also checked out.

If the check on the program certificate and the digital signature produces a positive result, then the user unit XSi further checks whether the proposed cryptographic algorithms that were proposed in the first negotiation communication can continue to be used to secure the transmission.

When the user unit XSi cannot support the proposed cryptographic algorithms, then the user unit, in other words, the server XSi, formulates a second proposal in a second proposal communication and transmits it to the multiplexer component ASC. The second proposal displays cryptographic processes that are supported by the user unit XSi. These [processes] are now proposed to the multiplexer component ASC as cryptographic processes to be employed as the procedure is further pursued for this logical connection between the multiplexer component and the user unit XSi.

The second proposal communication has at least the following components:

- the user certificate of the particular server XSi,

- a second random number that was generated by the user unit XSi itself,

- the second proposal,

- a digital signature that in each case would be formed at least via the first random number, the second random number as well as the second proposal.

The second proposal communication is transmitted to the multiplexer component ASC.

If the cryptographic algorithms given in the first proposal are supported by the user unit XSi, then the user unit XSi

formulates a confirmation communication and sends it to the multiplexer component ASC.

The confirmation communication has at least the following components:

- the user certificate,
- the second random number,
- a positive confirmation and
- a digital signature formed in each case at least via the first random number, the second random number and the positive confirmation.

The confirmation communication is sent to the multiplexer component ASC.

The multiplexer component ASC receives the negotiation communication or the confirmation communication and the multiplexer component ASC checks to see whether the user certificate as well as the digital signature are correct.

Furthermore, if the check yields a positive result and if the received communication was the confirmation communication, the multiplexer component ASC will generate a first session code, considering the agreed-upon cryptographic algorithms for a subsequent useful data transmission phase.

A first session code communication is formed from the first session code and is sent to the user unit XSi, which at least has the following components:

- the first session code that is encoded with a public code of the server XSi,

- a specification of the cryptographic processes to be employed,

- a digital signature that is formed at least via the first random number, the second random number, the first session code as well as the specification of the cryptographic processes to be employed.

If the multiplexer component ASC received the second negotiation communication and if the check on the user certificate and the digital signature or the hash value of the second negotiation communication yielded a positive result, then the multiplexer component ASC checks to see whether the cryptographic algorithms proposed in the second negotiation communication are supported by the multiplexer component ASC for the implementation of additional cryptographic processes.

When the proposed cryptographic processes are supported by the multiplexer component ASC, then a first session key is generated, considering the agreed-upon cryptographic algorithms for the following useful data transmission phase.

Furthermore, as described above, a first session code communication is sent to the multiplexer component ASC using the first session code.

This above-described procedure for "negotiating" the cryptographic processes that are to be employed is repeated until such time as both the user unit XSi and the multiplexer component ASC accept the last-proposed cryptographic processes.

In user unit XSi, the first session code is determined, employing a private code of the user unit XSi. Furthermore, the digital signature of the first session code communication is checked out.

Besides, if the check of the digital signature supplies a positive result, a second session code communication is formulated, employing a second session code that is formulated by the user unit XSi.

The second session code communication displays at least the following components:

- the second session code encoded with a public program code of the multiplexer component ASC and

- a digital signature that is formed at least via the first random number, the second random number, the second session code or a hash value formed via the same components. /7

The multiplexer component ASC receives the second session code communication and determines the second session code. The digital signature or the hash value of the second session code communication is now checked out.

If the check on the digital signature yielded a positive result, then the exchange session keys are employed in the following useful data transmission phase for the purpose of encoding the useful data. Here, each participating instance employs the session key that it generated itself for transmitting useful data, while the received session key is employed exclusively to receive useful data.

Other cryptographic processes for code exchange or to form the session code for useful data encoding can be employed in the context of the invention-based process without any restrictions.

The invention-based processes can be employed very advantageously in the following scenario.

Highly confidential information are exchanged between cross-linked computers in many private networks. Here the private network itself is mostly very well secured against the outside world, for example, by so-called Firewalls [6].

If a computer connected to the secured private network would like to communicate with a computer that can be reached outside that network only via an unsecured channel, for example, a computer that can be reached only via the Internet IN, then, so far, there was a big problem: No secure communication is possible in the Client-Server window systems based on [1].

In particular, one encounters the following problem: Other applications can be attacked via the window service program. To

prevent snooping on in-house information, company networks as a rule are not allowed to operate a window service program outside the company network. This generally customary restrictions, in particular, hinders synchronous CSCW systems that are based on Application Sharing.

These problems are described in detail, for example, in [6], [7].

These problems need not necessarily involve a communication that overlaps a local network; instead, this, for example, can also involve a secure Corporate Network CN where a communication partner wants to communicate with another communication partner of another company via the computer, for example, in a CSCW system.

By means of the invention-based processes, it is now possible when these processes are employed in a Firewall SC1, SC2 of the local network or of the Corporate Networks CN [sic; verb missing in original], whereby precisely the Firewall in each case is to be considered as a first securing computer unit SC1 or as a second securing computer unit SC2 (see Fig. 3).

The following publications were cited in this document:

[1] R. Scheifler et al., "The X Window System," ACM Transactions on Graphics, Vol. 5, № 2, pp. 79 to 109, April 1986.

[2] S. Garfinkel et al., "Practical UNIX Security," O'Reilly & Associates, Inc., ISBN 0-937175-72-2, pp. 221-253, 1991.

[3] H. Abdel-Wahab et al., "Issues, Problems and Solutions in Sharing X Clients on Multiple Displays," Internetworking: Research and Experience, Vol. 5, pp. 1 to 15, 1994.

[6] [sic; [4]] D. Garfinkel at al., "HP Shared X: A Tool for Real-Time Collaboration," Hewlett-Packard Journal, pp. 23 to 36, April 1994.

[5] J. Baldeschwieler et al., "A Survey of X Protocol Multiplexors," Swiss Federal Institute of Technology, Computer Engineering and Networks Laboratory (TIK), ETH-Zentrum, Zurich, 1993.

[6] S. Bellovin et al., "Network Firewalls," IEEE Communications Magazine, pp. 60 to 57 [sic], September 1994.

[7] G. Treese et al., "X Through the Firewall and Other Application Relays," Summer Usenix, 1993, 21 to 25, June, Cincinnati, pp. 87 to 98, 1993.

Claims

1. Process for the cryptographic securing of computer-assisted digital communication between a program (P) and at least one user unit (XSi),

- where program (P) forms (601) a communication (B),

- where a computer unit on which the program (P) is processed or a first securing computer unit (SC1) encodes (602) the communication (B) with a transport protocol (CB),

- where the encoded communication (CB) is decoded (603) using the transport protocol (DB),

- where the decoded communication (DB) is subjected (604) to a cryptographic process (VB),

- where the cryptographically processed communication (VB) is encoded (605) with the transport protocol (CVB) and

- where the encoded cryptographically processed communication (CVB) is transmitted (606) to the at least one user unit (XSi).

2. Process according to Claim 1,

- where the encoded cryptographically processed communication (CVB) is received (701) by the at least one user unit (XSi) or by a second securing computer unit (SC2),

- where using the transport protocol the encoded cryptographically processed communication (CVB) is decoded (702) (DVB),

- where the decoded cryptographically processed communication (DVB) is subjected (703) to a cryptographic processing (DEB) that is inverse with respect to the cryptographic process,

- where the inversely cryptographically processed communication (DEB) is encoded (704) with the transport protocol (CEB), and

- where the encoded inversely cryptographically processed communication (CEB) is decoded (705) using the transport protocol.

3. Process according to Claim 1, /8

- where the encoded cryptographically processed communication (CVB) is received by the at least one user unit (XSi),

- where, using the transport protocol, the encoded cryptographically processed communication (CVB) is decoded (DVB) and

- where the decoded cryptographically processed communication (DVB) is subjected to a cryptographic processing (DEB) that is inverse with respect to the cryptographic process.

4. Process for the cryptographic securing of computer-assisted digital communication between a program (P) and at least one user unit (XSi),

- where a user unit (XSi) formulates (901) a request (A),

- where the user unit (XSi) or a second securing computer unit (SC2) encodes (902) the request (A) with a transport protocol,

- where the encoded request (CA) is decoded (903) using the transport protocol (DA),

- where the decoded request (DA) is subjected (904) to a cryptographic process (VA),

- where the cryptographically processed request (VA) is encoded (905) with the transport protocol (CVA) and

- where the encoded cryptographically processed request (CVA) is transferred (906) to the program (P).

5. Process according to Claim 4,

- where the encoded cryptographically processed request (CVA) is received (1001) by the program (P) or by a first securing computer unit (SC1),

- where, using the transport protocol, the encoded cryptographically processed request (CVA) is decoded (1002) (DVA),

- where the decoded cryptographically processed request (DVA) is subjected (1003) to a cryptographic processing (DEA) that is inverse with respect to the cryptographic process - where the inversely cryptographically processed request (DEA) is encoded (1004) with the transport protocol (CEA) and

- where the encoded inversely cryptographically processed request (CEA) is decoded (1005) using the transport protocol.

6. Process according to Claim 4,

- where the encoded cryptographically processed request (CVA) is received by the program (P),

- where, using the transport protocol, the encoded cryptographically processed request (CVA) is decoded (DVA) and

- where the decoded cryptographically processed request (DVA) is subjected to a cryptographic processing (DEA) that is inverse with respect to the cryptographic process.

7. Process for cryptographic securing of computer-assisted digital communication between a program (P) and at least one user unit (XSi),

- where program (P) formulates (1101) a communication (B),

- where the communication (B) is subjected (1102) to a cryptographic process (VB),

- where the cryptographically processed communication (VB) is encoded (1103) with the transport protocol (CVB),

- where the encoded cryptographically processed communication (CVB) is transmitted (1004) to at least one user unit (XSi),

- where the encoded cryptographically processed communication (CVB) is received (1105) by at least one user unit (XSi) or of a second securing computer unit (SC2),

- where, using the transport protocol, the encoded cryptographically processed communication (CVB) is decoded (1106) (DVB),

- where the decoded cryptographically processed communication (DVB) is subjected (1107) to a cryptographic processing (DEB) that is inverse with respect to the cryptographic process,

- where the inversely cryptographically processed communication (DEB) is encoded (1108) using the transport protocol (CEB) and

- where the encoded inversely cryptographically processed communication (CEB) is decoded (1109) using the transport protocol.

8. Process for the cryptographic securing of computer-assisted digital communication between a program (P) and at least one user unit (XSi),

- where at least one user unit (XSi) formulates (1201) a request (A),

- where the decoded request (DA0 is subjected (1202) to a cryptographic process (VA),

- where the user unit (XSi) or a second securing computer unit (SC2) encodes (1203) the cryptographically processed request (A) with a transport protocol (CA),

- where the encoded cryptographically processed request (CVA) is transferred (1204) to the program (P),

- where the encoded cryptographically processed request (CVA) is received (1205) by the program (P) or by a first securing computer unit (SC1),

- where, using the transport protocol, the encoded cryptographically processed request (CVA) is decoded (1206) (DVA),

- where the decoded cryptographically processed request (DVA) is subjected (1207) to a cryptographic processing (DEA) that is inverse with respect to the cryptographic process,

- where the inversely cryptographically processed request (DEA) is encoded (1208) with the transport protocol (CEA) and /9

- where the encoded inversely cryptographically processed request (CEA) is decoded (1209) using the transport protocol.

9. Process according to one of Claims 1 to 8,

- where the cryptographic processing is performed (81) at least by one encoding of the request (Ai) and

- where the inverse cryptographic processing is carried out at least by one decoding of the request (Ai).

10. Process according to one of Claims 1 to 9,

- where the cryptographic processing is carried out (82) at least by means of authentication mechanisms for the request (Ai) and

- where the inverse cryptographic processing is carried out at least by inverse authentication mechanisms for the request (Ai).

11. Process according to one of Claims 1 to 10,

- where the cryptographic processing is carried out (83) at least by access control mechanisms for request (Ai) and

- where the inverse cryptographic processing is carried out at least by inverse access control mechanisms for request (Ai).

12. Process according to one of Claims 1 to 11, where at the start of the process, a cryptographic initialization phase with formation of a session code is performed for each connection of a user unit (XSi) with the program (P).

13 pages of drawings.

[Please insert Fig. 6].

[Key: 601) Communication B is formulated by program P; 602) Communication is encoded in transport protocol layer TP; 603) Communication is decoded in transport protocol layer TP; 604) Communication is subjected to cryptographic processes; 605) Communication is encoded in transport protocol layer TP; 606) Encoded communication is transferred to user unit XS].

[Please insert Fig. 8].

[Key: 80) Cryptographic process in security layer SL; 81) Encoding; 82) Authentication mechanisms; 83) Access control mechanisms].

[Please insert Fig. 7].

[Key: 601) Communication B is formulated by program P; 602) Communication is encoded in transport protocol layer TP; 603) Communication is decoded in transport protocol layer TP; 604) Communication is subjected to cryptographic processes; 605) Communication is encoded in transport protocol layer TP; 606) Encoded communication is transferred to user unit XS; 701) Encoded communication is received by user unit XS; 702) Received communication is decoded in transport layer TP; 703) Decoded communication is processed in an inversely cryptographic manner in security layer SL; 704) Inversely processed communication is encoded in transport layer TP; 705) Encoded communication is decoded in transport layer TP].

[Please insert Fig. 9].

[Key: 901) Request A is formulated by user unit XS; 902) Request A is encoded in transport protocol layer TP; 903) Request A is decoded in transport protocol layer TP; 904) Request A is subjected to cryptographic processes; 905) Request A is encoded in transport protocol layer TP; 906) Encoded request is transferred to program P].

[Please insert Fig. 10].

[Key: 901) Request A is formulated by user unit XS; 902) Request A is encoded in transport protocol layer TP; 903) Request A is decoded in transport protocol layer TP; 904) Request A is

subjected to cryptographic processes; 905) Request A is encoded in transport protocol layer TP; 906) Encoded request A is transferred to program P; 1001) Encoded request A is received by program P; 1002) Received request A is decoded in transport layer TP; 1003) Decoded request A is processed in an inversely cryptographic manner in security layer SL; 1004) Inversely processed request A is encoded in transport layer TP; 1005) Encoded request A is decoded in transport layer TP].

[Please insert Fig. 11].

[Key: 1101) Communication B is formulated by program P; 1102) Communication B is subjected to cryptographic processed; 1103) Communication B is encoded in transport protocol layer TP; 1104) Encoded communication B is transferred to user unit XS; 1105) Encoded communication B is received by user unit XS; 1106) Received communication B is decoded in transport layer TP; 1107) Decoded communication B is processed in an inversely cryptographic manner in security layer SL; 1108) Communication B is encoded in transport protocol layer TP; 1109) Communication B is decoded in transport protocol layer TP].

[Please insert Fig. 12].

[Key: 1201) Request A is formulated by user unit XS; 1202) Request A is subjected to cryptographic processed; 1203) Request A is encoded in transport protocol layer TP; 1204) Encoded request A is transferred to program P; 1205) Encoded request A

is received by program P; 1206) Received request A is decoded in transport layer TP; 1207) Decoded request A is processed in an inversely cryptographic manner in security layer SL; 1208) Request A is encoded in transport protocol layer TP; 1209) Request A is decoded in transport protocol layer TP].

FIG 1

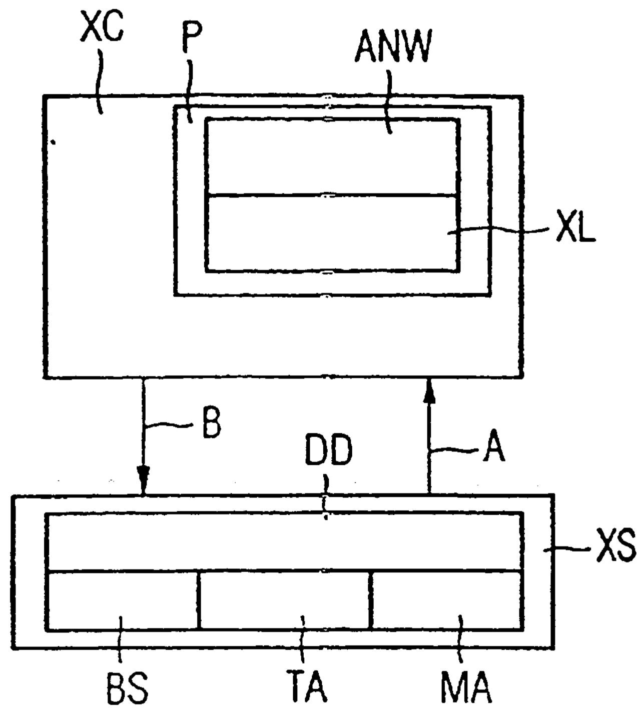

XC
P
ANW
XL
B
DD
A
XS
BS
TA
MA

FIG 2

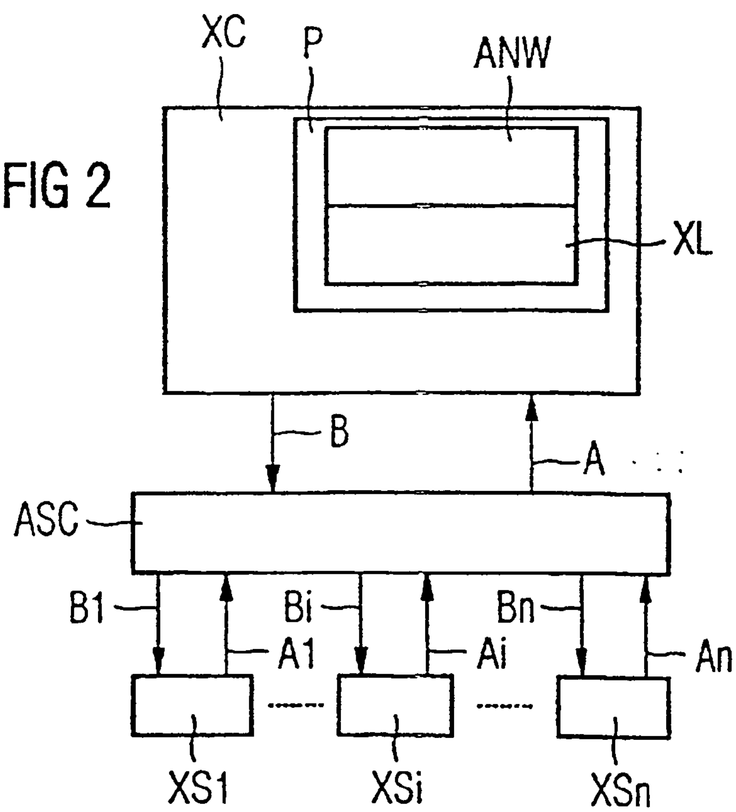

XC
P
ANW
XL
B
A
ASC
B1
A1
Bi
Ai
Bn
An
XS1
XSi
XSn

FIG 3

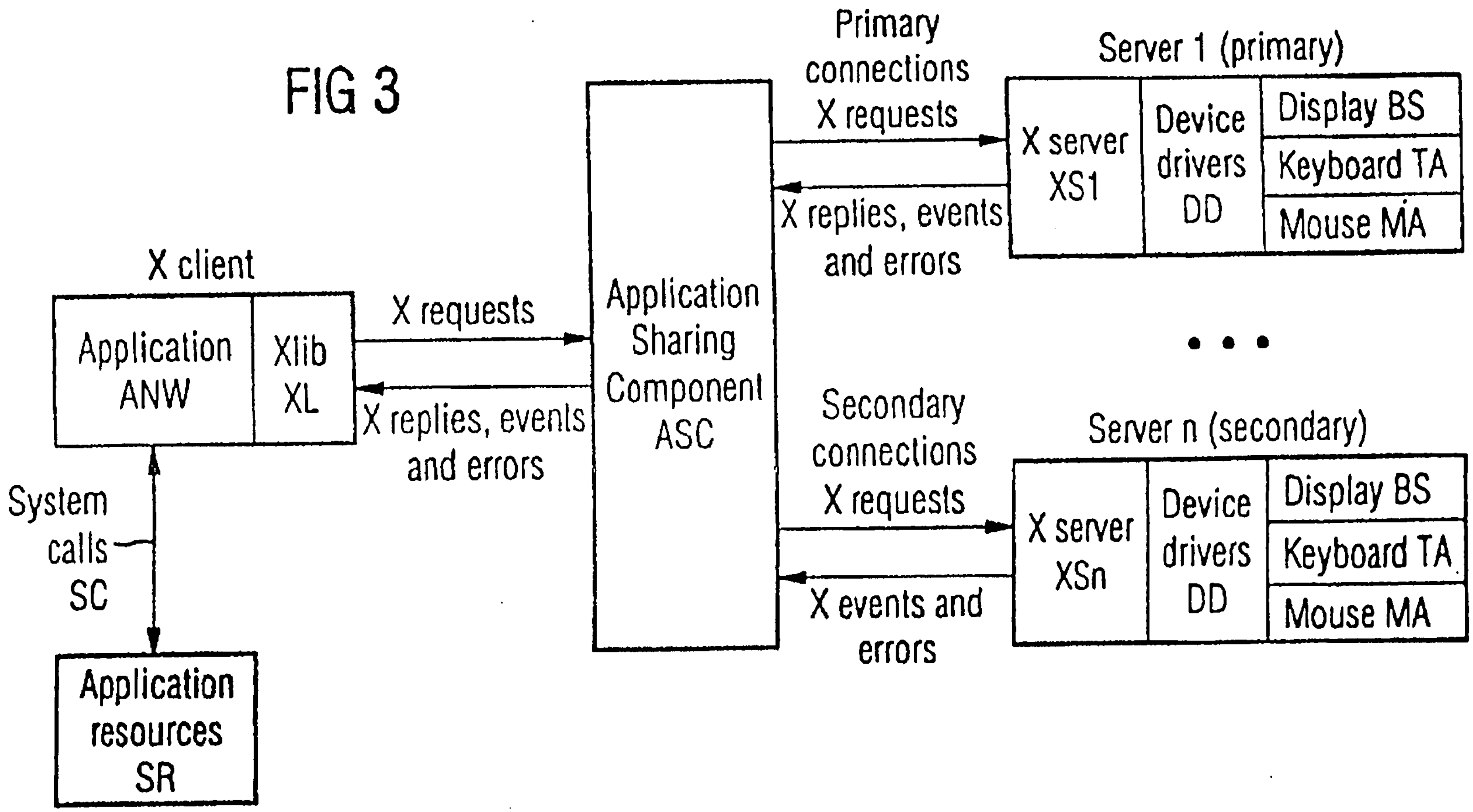
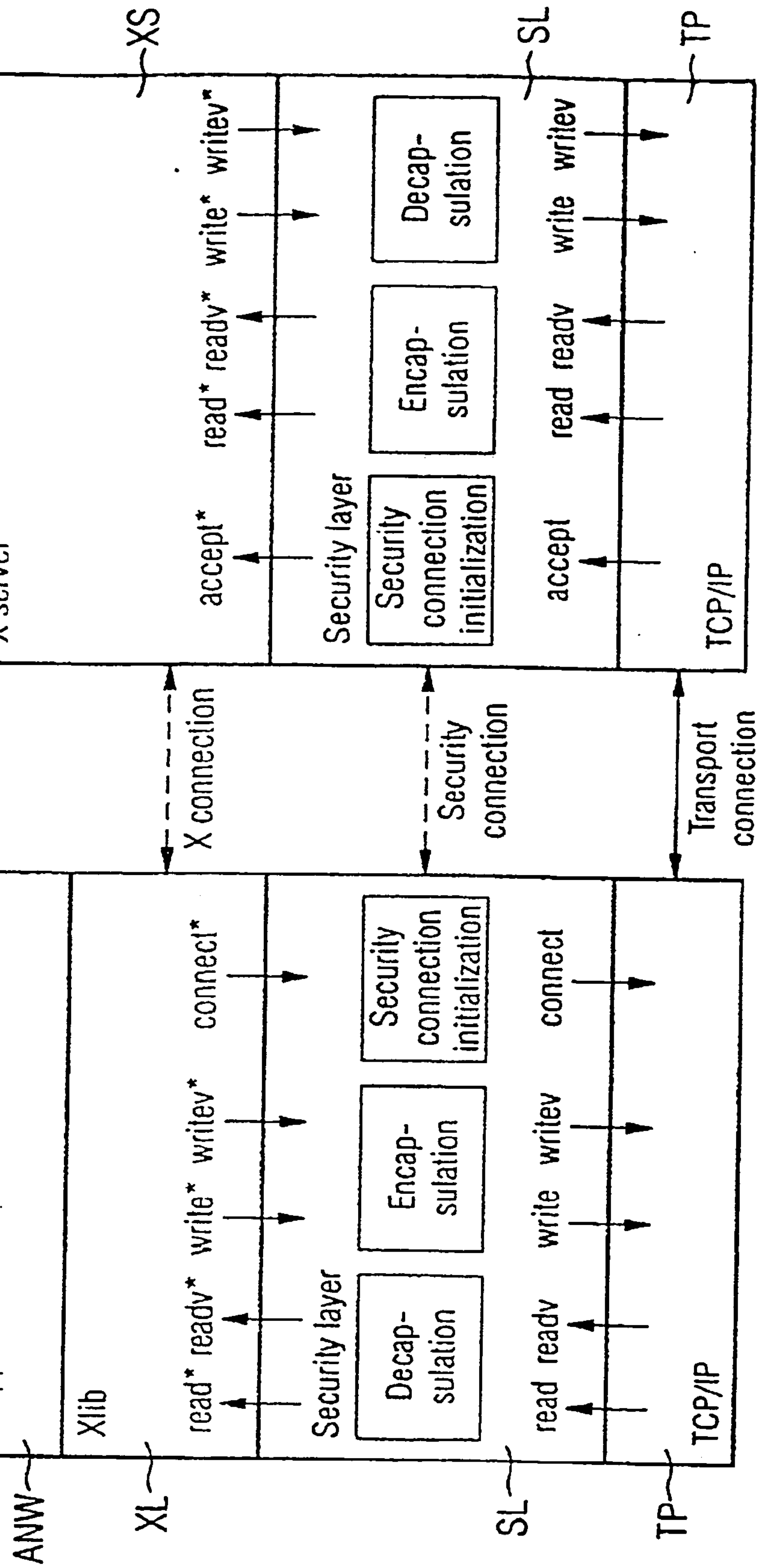
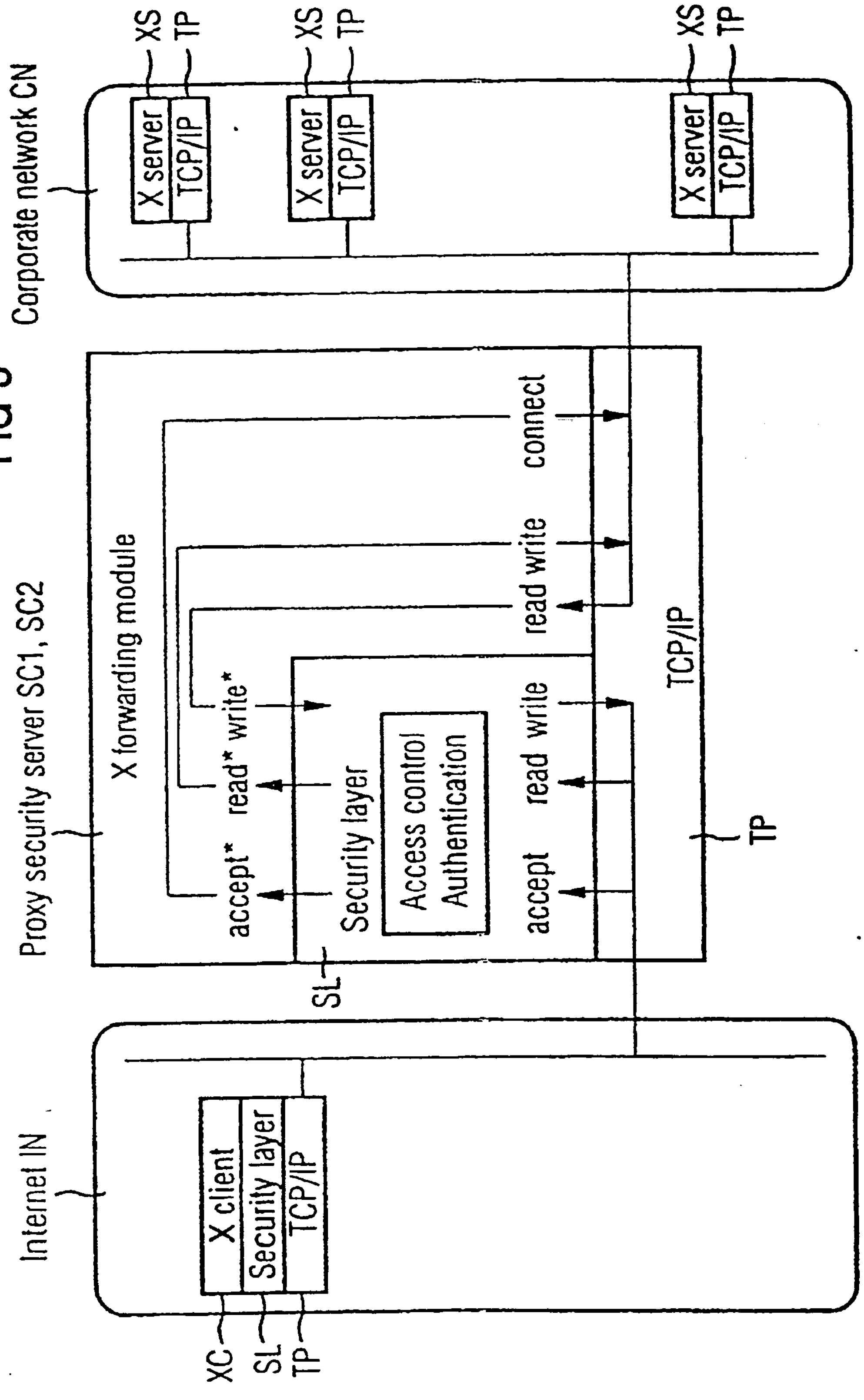

X client
Application ANW
Xlib XL
X requests
X replies, events and errors
System calls SC
Application resources SR
Application Sharing Component ASC
Primary connections X requests
X replies, events and errors
Server 1 (primary)
X server XS1
Device drivers DD
Display BS
Keyboard TA
Mouse MA
Secondary connections X requests
X events and errors
Server n (secondary)
X server XSn
Device drivers DD
Display BS
Keyboard TA
Mouse MA

FIG 4



X application
Xlib
read* readv* write* writev* connect*
Security layer
Decap-
sulation
Encap-
sulation
Security
connection
initialization
read readv write writev connect
TCP/IP
ANW
XL
SL
TP
X connection
Security
connection
Transport
connection
X server
accept* read* readv* write* writev*
Security layer
Security
connection
initialization
Encap-
sulation
Decap-
sulation
accept read readv write writev
TCP/IP
XS
SL
TP

FIG 5



Internet IN
X client
Security layer
TCP/IP
XC
SL
TP
Proxy security server SC1, SC2
X forwarding module
accept*
read*
write*
Security layer
Access control
Authentication
accept
read
write
read
write
read
write
connect
TCP/IP
TP
SL
Corporate network CN
X server
TCP/IP
XS
TP
X server
TCP/IP
XS
TP
X server
TCP/IP
XS
TP

FIG 6

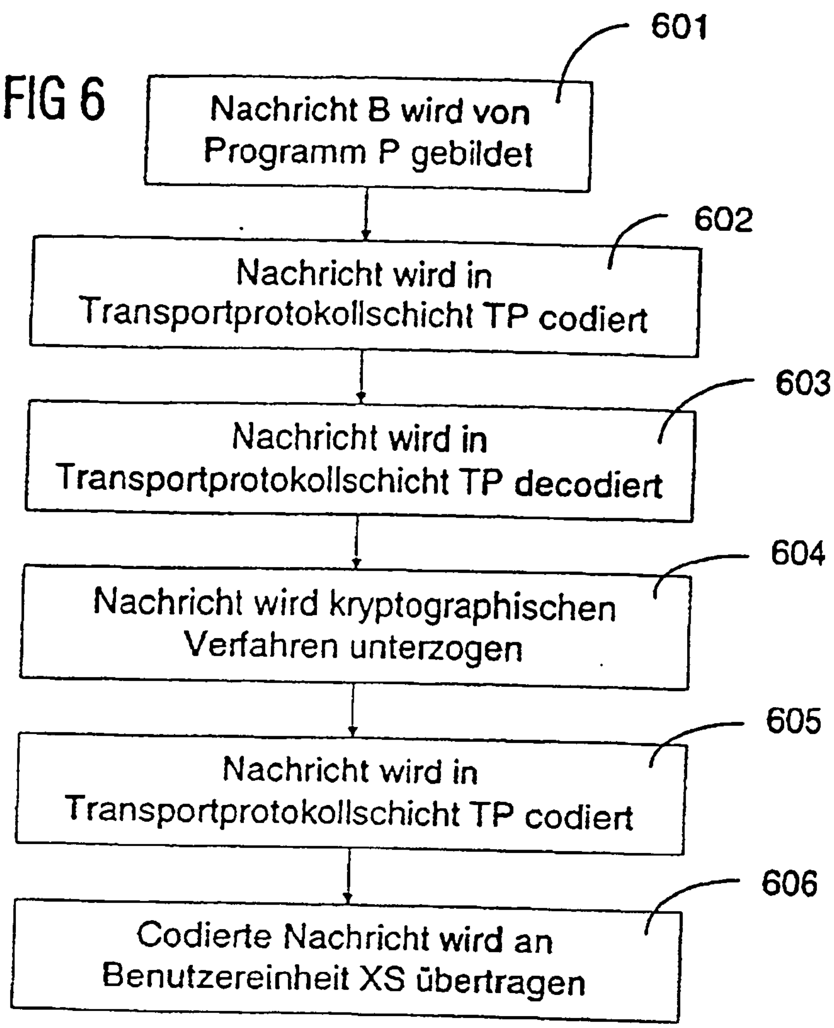

601
Nachricht B wird von Programm P gebildet
602
Nachricht wird in Transportprotokollschicht TP codiert
603
Nachricht wird in Transportprotokollschicht TP decodiert
604
Nachricht wird kryptographischen Verfahren unterzogen
605
Nachricht wird in Transportprotokollschicht TP codiert
606
Codierte Nachricht wird an Benutzereinheit XS übertragen

FIG 8

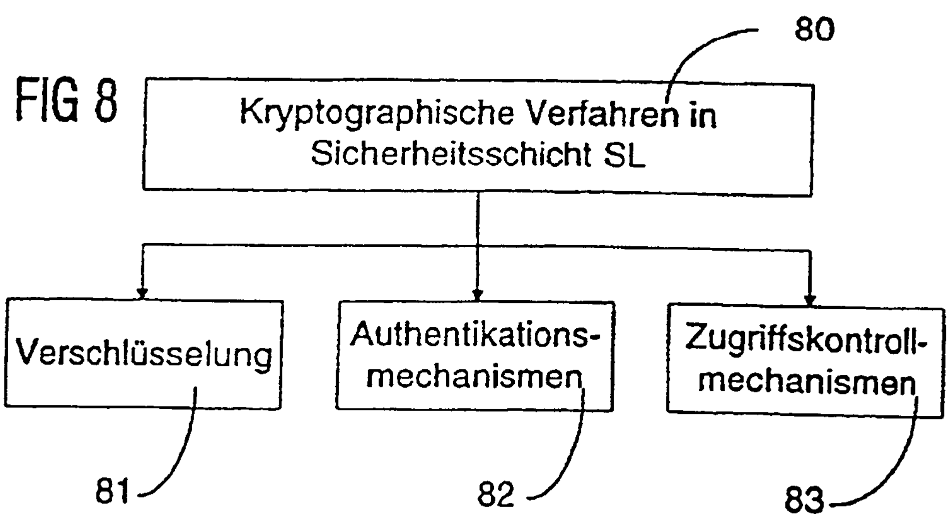

80
Kryptographische Verfahren in Sicherheitsschicht SL
Verschlüsselung
Authentikations-mechanismen
Zugriffskontroll-mechanismen
81
82
83

FIG 7

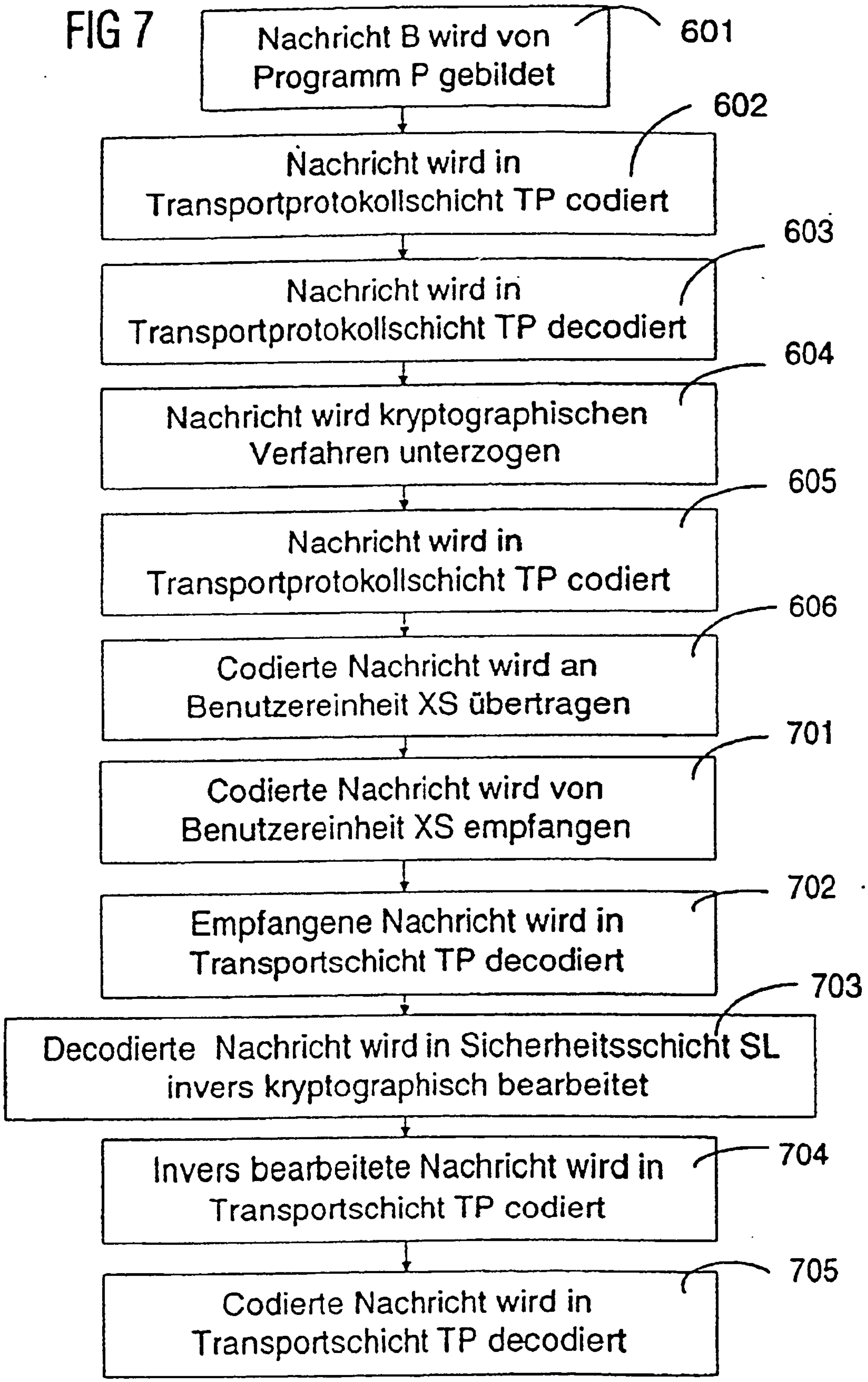

Nachricht B wird von Programm P gebildet
601
Nachricht wird in Transportprotokollschicht TP codiert
602
Nachricht wird in Transportprotokollschicht TP decodiert
603
Nachricht wird kryptographischen Verfahren unterzogen
604
Nachricht wird in Transportprotokollschicht TP codiert
605
Codierte Nachricht wird an Benutzereinheit XS übertragen
606
Codierte Nachricht wird von Benutzereinheit XS empfangen
701
Empfangene Nachricht wird in Transportschicht TP decodiert
702
Decodierte Nachricht wird in Sicherheitsschicht SL invers kryptographisch bearbeitet
703
Invers bearbeitete Nachricht wird in Transportschicht TP codiert
704
Codierte Nachricht wird in Transportschicht TP decodiert
705

FIG 9

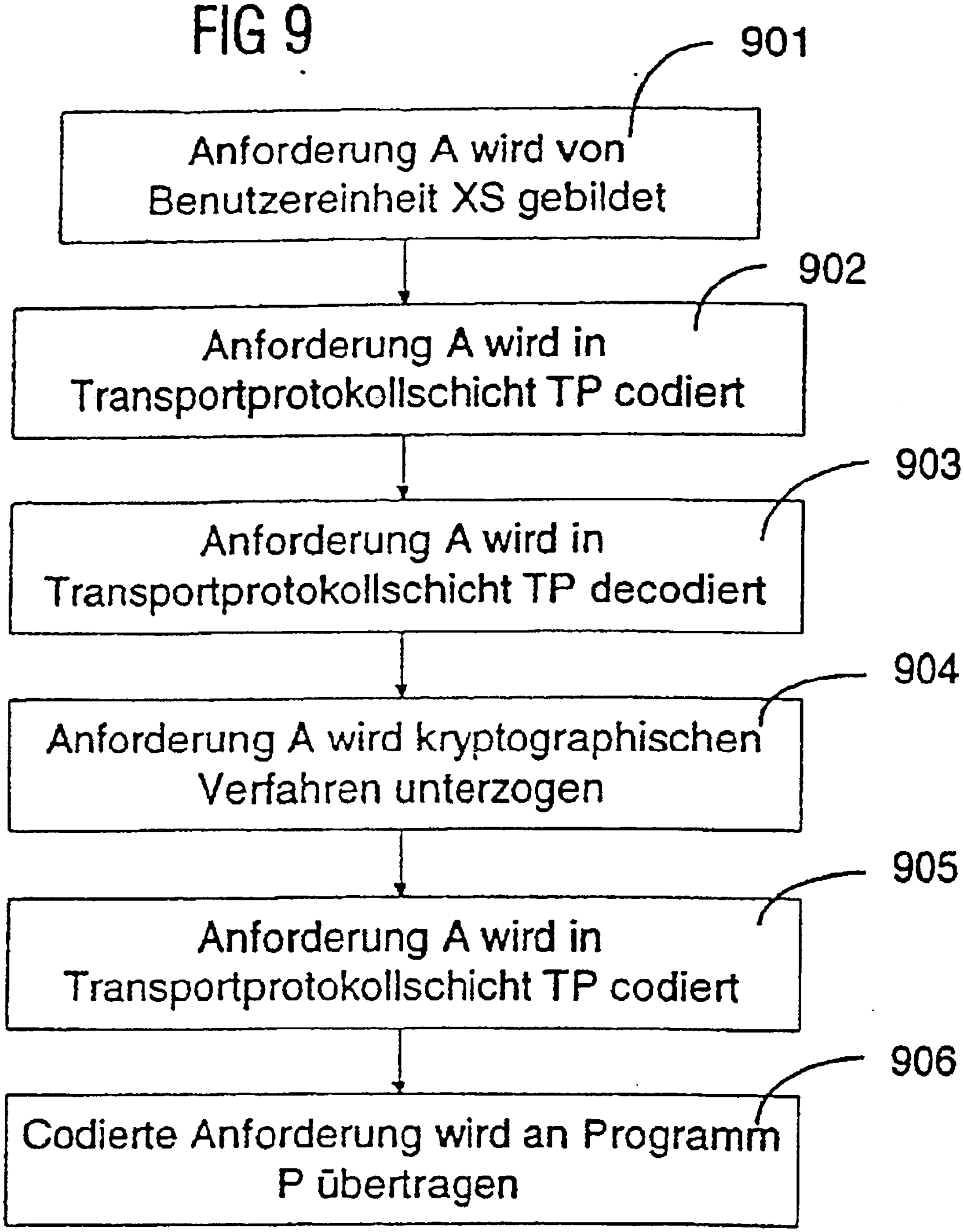
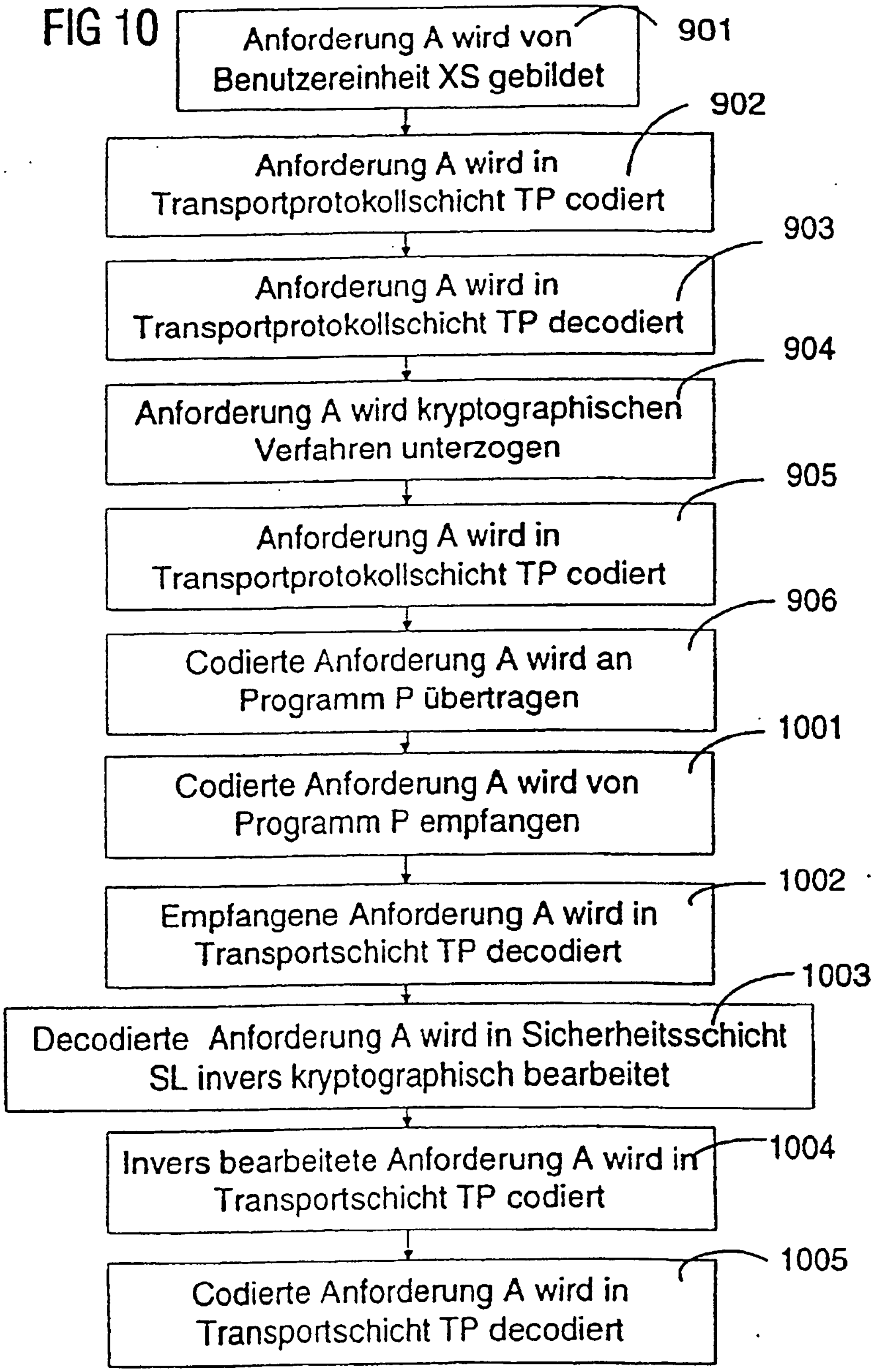

901
Anforderung A wird von Benutzereinheit XS gebildet
902
Anforderung A wird in Transportprotokollschicht TP codiert
903
Anforderung A wird in Transportprotokollschicht TP decodiert
904
Anforderung A wird kryptographischen Verfahren unterzogen
905
Anforderung A wird in Transportprotokollschicht TP codiert
906
Codierte Anforderung wird an Programm P übertragen

FIG 10



Anforderung A wird von Benutzereinheit XS gebildet
901
Anforderung A wird in Transportprotokollschicht TP codiert
902
Anforderung A wird in Transportprotokollschicht TP decodiert
903
Anforderung A wird kryptographischen Verfahren unterzogen
904
Anforderung A wird in Transportprotokollschicht TP codiert
905
Codierte Anforderung A wird an Programm P übertragen
906
Codierte Anforderung A wird von Programm P empfangen
1001
Empfangene Anforderung A wird in Transportschicht TP decodiert
1002
Decodierte Anforderung A wird in Sicherheitsschicht SL invers kryptographisch bearbeitet
1003
Invers bearbeitete Anforderung A wird in Transportschicht TP codiert
1004
Codierte Anforderung A wird in Transportschicht TP decodiert
1005

FIG 11

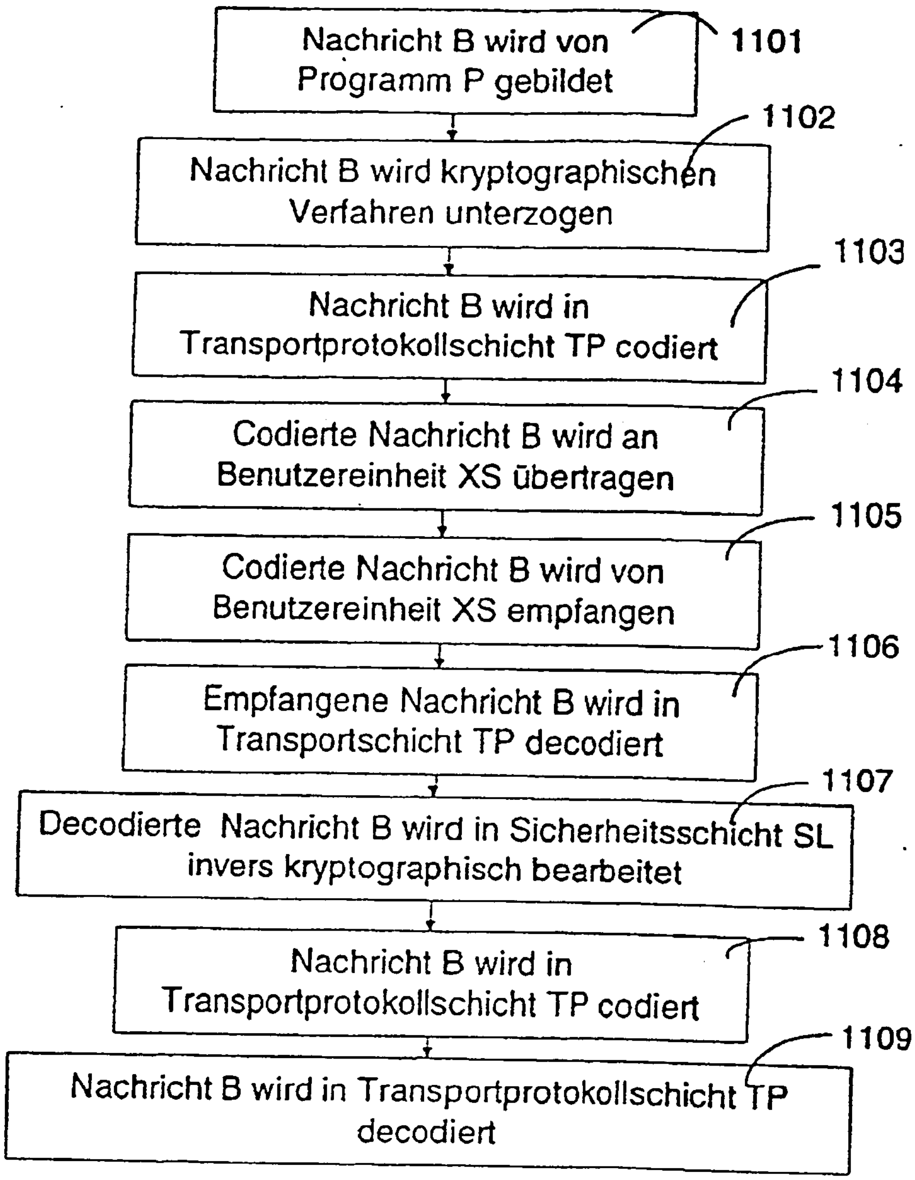
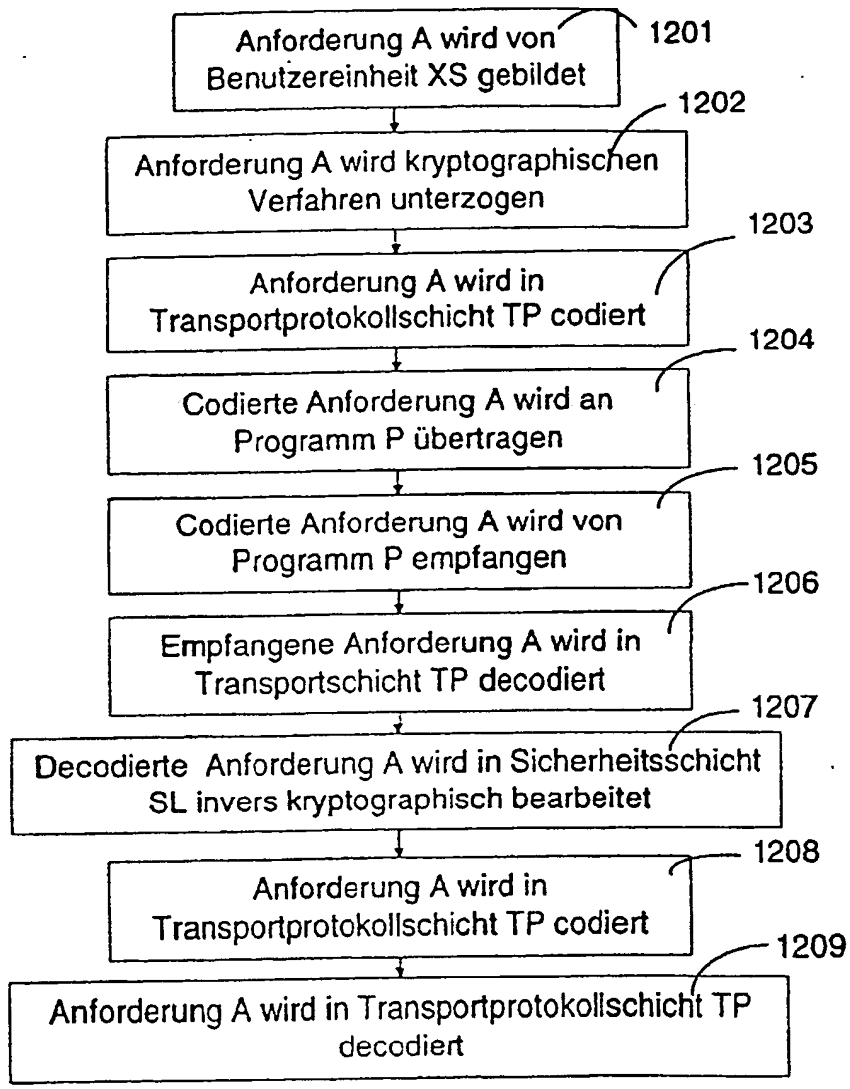

Nachricht B wird von Programm P gebildet
1101
Nachricht B wird kryptographischen Verfahren unterzogen
1102
Nachricht B wird in Transportprotokollschicht TP codiert
1103
Codierte Nachricht B wird an Benutzereinheit XS übertragen
1104
Codierte Nachricht B wird von Benutzereinheit XS empfangen
1105
Empfangene Nachricht B wird in Transportschicht TP decodiert
1106
Decodierte Nachricht B wird in Sicherheitsschicht SL invers kryptographisch bearbeitet
1107
Nachricht B wird in Transportprotokollschicht TP codiert
1108
Nachricht B wird in Transportprotokollschicht TP decodiert
1109

FIG 12



Anforderung A wird von Benutzereinheit XS gebildet
1201
Anforderung A wird kryptographischen Verfahren unterzogen
1202
Anforderung A wird in Transportprotokollschicht TP codiert
1203
Codierte Anforderung A wird an Programm P übertragen
1204
Codierte Anforderung A wird von Programm P empfangen
1205
Empfangene Anforderung A wird in Transportschicht TP decodiert
1206
Decodierte Anforderung A wird in Sicherheitsschicht SL invers kryptographisch bearbeitet
1207
Anforderung A wird in Transportprotokollschicht TP codiert
1208
Anforderung A wird in Transportprotokollschicht TP decodiert
1209

FIG 13

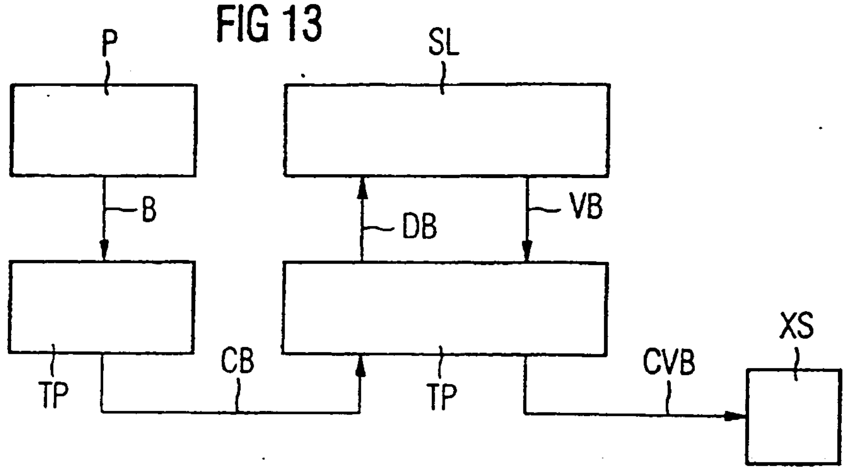


FIG 14

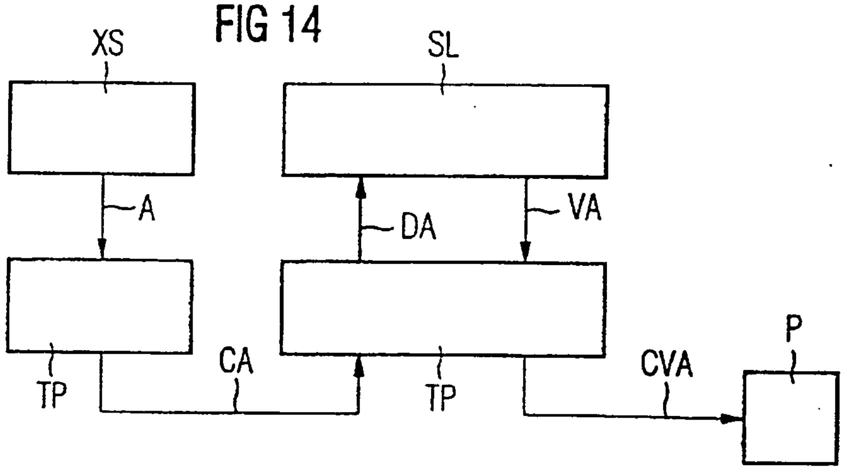


FIG 15

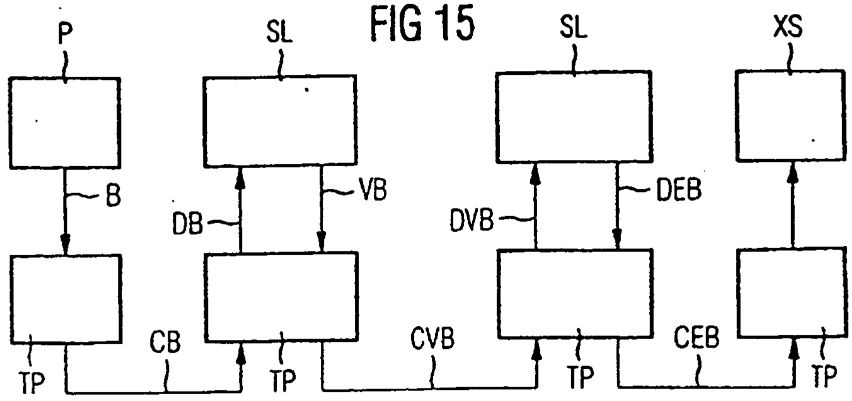
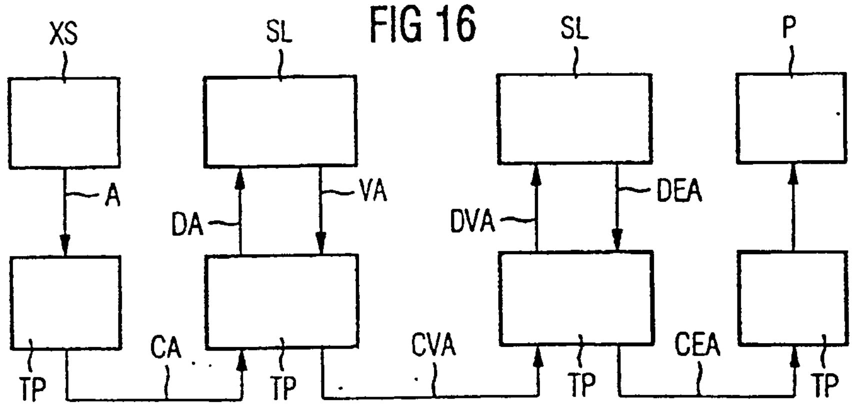

P
SL
SL
XS
B
DB
VB
DVB
DEB
CB
CVB
CEB
TP
TP
TP
TP

FIG 16


XS
SL
SL
P
A
DA
VA
DVA
DEA
CA
CVA
CEA
TP
TP
TP
TP

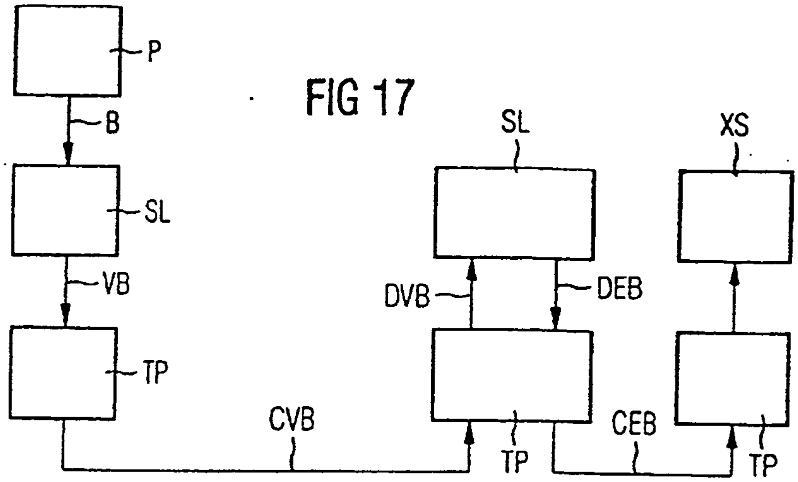

P
B
SL
VB
TP
CVB
SL
XS
DVB
DEB
TP
CEB
TP

FIG 17

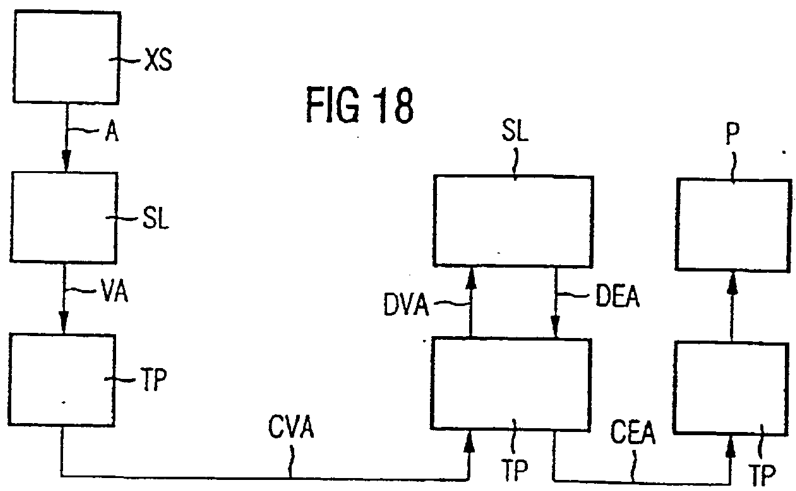

XS
A
SL
VA
TP
CVA
SL
P
DVA
DEA
TP
CEA
TP

FIG 18